

⑨ 日本国特許庁(JP)

⑩ 特許出願公開

⑫ 公開特許公報(A) 平4-54638

⑬ Int. Cl.⁵

G 06 F 9/38

識別記号

3 7 0 C
3 8 0 A

庁内整理番号

7927-5B
7927-5B

⑬ 公開 平成4年(1992)2月21日

審査請求 未請求 請求項の数 2 (全5頁)

⑭ 発明の名称 電子計算機

⑮ 特 願 平2-164552

⑯ 出 願 平2(1990)6月22日

⑰ 発 明 者 皆 川 健 二 神奈川県川崎市幸区小向東芝町1番地 株式会社東芝総合
研究所内

⑰ 発 明 者 相 川 健 神奈川県川崎市幸区小向東芝町1番地 株式会社東芝総合
研究所内

⑰ 発 明 者 齊 藤 光 男 神奈川県川崎市幸区小向東芝町1番地 株式会社東芝総合
研究所内

⑰ 出 願 人 株 式 会 社 東 芝 神奈川県川崎市幸区堀川町72番地

⑰ 代 理 人 弁 理 士 鈴 江 武 彦 外3名

明 細 書

1. 発明の名称

電 子 計 算 機

2. 特許請求の範囲

(1) 複数の命令を同時にフェッチするフェッチ回路と、このフェッチ回路にフェッチされた複数の命令を複数の実行ユニットにそれぞれ分配供給して並列的に実行させる命令分配器とを具備した電子計算機において、

前記命令分配器による分配供給処理と並行して、前記フェッチ回路にフェッチされた複数の命令をそれぞれデコードして上記各命令間の実行順序依存性を検出する第1の処理ユニットを具備したことを特徴とする電子計算機。

(2) 第1の処理ユニットにより検出された複数の命令間の実行順序依存性に従って、前記ユニットに供給された命令をアボート制御する第2の処理ユニットを備えたことを特徴とする請求項(1)に記載の電子計算機。

3. 発明の詳細な説明

[発明の目的]

(産業上の利用分野)

本発明は複数の命令を並列的に実行する電子計算機に関する。

(従来の技術)

近時、複数の命令を細粒度レベルで並列的に実行してその演算処理速度の高速化を図るようにした並列処理型の電子計算機の開発が種々試みられている。特に最近では、例えばマルチフローコンピュータ社の【トレース】やインテル社の【I860】に代表されるような、VLIW方式と称される並列処理型の電子計算機が開発され、注目されている。然し乍ら、この種の従来方式の電子計算機にあっては、従来一般的なシーケンシャルな命令コードを実行することができないと云う本質的な問題がある。

また従来のシーケンシャルな命令コードを、そのオブジェクトの互換性を保ちながら並列的に実行するには、シーケンシャルに与えられる複数の

命令の同時実行可能性を検定することが必要である。しかしこのような検定を行うには、例えば1サイクルのロジック段数を増やすことが必要であり、この結果、そのサイクルタイムが増大することが否めない。

一方、レーテンシのある命令の実行をサポートする為に複数の命令間の依存性解析を行う手法としてレジスタスコアボード方式が知られている。このレジスタスコアボード方式は、例えばモトローラの【M88000】に見られるように、インストラクションバッファにフェッチした命令をレジスタフィールド・デコーダにて解析（デコード）し、この解析処理によって求められる上記命令のソースレジスタ番号、ディスティネーションレジスタ番号をレジスタファイルに与えると共に、レジスタスコアボード回路に与える。そしてこのレジスタスコアボード回路にて複数の命令間の依存性を解析し、その解析結果に従って前記命令の実行手順を制御するようにしたものである。

そこでこのようなレジスタスコアボード方式を

利用し、複数の命令実行ユニットを用いて並列的に処理しようとする複数の命令間の依存性から、それらの同時実行可能性を解析することが考えられる。例えばインストラクションバッファに、連続した複数の命令をフェッチし、命令分配器にてこれらの命令の種別をそれぞれ判定し、この判定結果に従ってその命令を実行するに適した実行ユニットを選定して上記各命令をそれぞれ分配供給して前記各命令を並列的に実行することが考えられている。

具体的には浮動小数点乗算ユニット、浮動小数点加算ユニット、算術演算ユニット（ALU）等の種々の命令実行ユニットを準備し、これらの複数の命令実行ユニットに対して、その命令実行機能に適した命令をそれぞれ分配供給して上記各命令を並列的に実行することが考えられている。

この場合、命令間の依存性の解析は、各命令実行ユニットにそれぞれ組み込まれたレジスタフィールド・デコーダとレジスタスコアボード回路とを用いて行われる。

ところがこのような命令間の依存性解析は、命令分配器を介して命令実行ユニットに命令が分配供給された後に各命令実行ユニットのレジスタスコアボード回路を用いて行われるので、その処理に長い時間が掛かると言う問題がある。つまり複数の命令の同時実行可能性を検定するのに長い処理時間を必要とすると言う問題がある。

（発明が解決しようとする課題）

このように従来一般的なシーケンシャルな命令コードを、そのオブジェクトの互換性を保ちながら並列的に処理しようとする場合には、複数の命令間の実行順序依存性を検出し、複数の命令の同時実行を制御する必要がある。しかし従来のレジスタスコアボード方式をそのまま用いて命令間の依存性を解析しようすると、命令実行ユニットに命令を分配供給した後、その命令実行ユニットにて命令間の依存性の解析が行われるので、その処理に多くの時間を必要とすると言う問題があった。

本発明はこのような事情を考慮してなされたも

ので、その目的とするところは、複数の命令間の実行順序依存性を高速に検出して複数の命令の並列的な処理を効率的に行うことの可能な電子計算機を提供することにある。

〔発明の構成〕

（課題を解決するための手段）

本発明は複数の命令を同時にフェッチするフェッチ回路と、このフェッチ回路にフェッチされた複数の命令を複数の実行ユニットにそれぞれ分配供給して並列的に実行させる命令分配器とを具備し、上記複数の実行ユニットにて複数の命令を並列的に実行し得るようにした電子計算機に係り、

前記命令分配器による命令の分配供給処理と並行して、前記フェッチ回路にフェッチされた複数の命令をデコードして複数の命令間の実行順序依存性を検出する第1の処理ユニットと、この第1の処理ユニットにより検出された複数の命令間の実行順序依存性に従って前記実行ユニットに供給された命令をアポート制御する第2の処理ユニットとを具備し、同時実行可能な順序で複数の実行

ユニットにそれぞれ命令を実行させるようにしたことを特徴とするものである。

(作用)

本発明によれば、フェッチ回路にフェッチされた複数の命令をその命令の種類に応じて複数の実行ユニットに分配供給し、各実行ユニットにて命令のデコードとその実行を行なわせる処理と並行して、第1の処理ユニットにて前記フェッチ回路にフェッチされた複数の命令を直接的にデコードして複数の命令間の実行順序依存性を検出する処理を同時に実行し、この検出結果に従って前記複数の実行ユニットにおける命令の実行が制御される。従って実行順序依存性の検出処理によって、例えば同時実行の可能性が否定される命令が検出されたような場合、速やかにその命令が分配された実行ユニットにおける命令の実行を中止制御することが可能となる。つまり命令が分配供給された個々の実行ユニットにて命令間の依存性を検出するような時間的な遅れを招くことなく、速早く同時実行不可能な命令を検出してその命令の実行

を効率的に中止させることができる。

(実施例)

以下、図面を参照して本発明の一実施例に係る電子計算機について説明する。

第1図は実施例に係る電子計算機の要部概略構成を示す図であり、1はシーケンシャルな命令を複数個ずつフェッチするインストラクション・バッファ(フェッチ回路)である。尚、ここでは4個の命令11,12,13,14を同時にフェッチするように構成されたインストラクション・バッファ1が示される。

しかしてこのインストラクション・バッファ1にフェッチされた4つの命令11,12,13,14は、命令分配器2を介して、その命令の種類に応じて複数の命令実行ユニット3(3a,3b,...3n)にそれぞれ分配供給される。この実施例では複数の命令実行ユニット3として、浮動小数点加算ユニット(ADD)3a、浮動小数点乗算ユニット(MUL)3b、算術演算ユニット(ALU)3c,3dがそれぞれ準備されている。前記命令分配器2は前記バッ

ファ1にフェッチされた命令が浮動小数点加算命令の場合には、その命令を上記浮動小数点加算ユニット(ADD)3aに供給し、また命令が浮動小数点乗算命令である場合には浮動小数点乗算ユニット(MUL)3bに、更に整数演算命令である場合には前記算術演算ユニット(ALU)3c,3dに対してその命令をそれぞれ分配供給する。

これらの各命令実行ユニット3はそれぞれデコード4を備えており、基本的には前記命令分配器2の制御の下で前記インストラクション・バッファ1から選択的に分配供給された命令をそれぞれデコードし、そのデコード結果が示すレジスタフィールドのデータに従ってレジスタをアクセスしてその命令を実行する。

一方、前記インストラクション・バッファ1にフェッチされた4つの命令11,12,13,14は命令分配器2に供給されて命令実行ユニット3に分配供給されると同時に、同時実行可能性検出部(第1の処理ユニット)5に与えられる。この同時実行可能性検出部5は前記インストラクション

・バッファ1から与えられる命令をそれぞれデコードするデコード5aと、そのデコード結果を検定して複数の命令間の同時実行の可能性を判断する検定部5bとを備えており、これらのデコード5aと検定部5bとを用いて前記インストラクション・バッファ1にフェッチされた4つの命令11,12,13,14間での同時実行可能性を判断する。

具体的にはこの複数の命令の同時実行可能性の判断は、例えば前述した命令11,12,...が

11 [ADD r3, r4, r5]

12 [ADD r6, r7, r8]

;

として与えられるような場合、例えば命令11を基準として次の命令12の実行条件を判断する。この場合には命令12の全てのレジスタフィールドが、命令11のディスティネーション・レジスタフィールドr3と異なることから、上記命令12は命令11と同時に実行可能であると判断する。このような判断処理は、先の命令に対して次の命令の実行条件が成立するか否かを次々と判断

することによりなされる。つまりこのような同時実行可能性の判断は、次の命令13については先の命令11,12に対して同様に行われ、更に次の命令14については命令11,12,13のそれぞれに対して同様に行われる。

しかして前述した命令11,12...が、例えば

11 [ADD r3, r4, r5]

12 [ADD r6, r7, r8]

⋮

として与えられるような場合には、第2図に示すように命令11を基準として次の命令12の実行条件を判断すると、命令12のソースレジスタフィールドr3が、命令11のディスティネーション・レジスタフィールドr3と一致することが検出される。この結果、前記同時実行可能性検出部5は、上記命令12については命令11の実行完了後に実行する必要がある、つまり命令12と命令11とは同時に実行することが不可能であると判断する。

アポート信号供給器(第2の処理ユニット)6

た命令との間でその同時実行可能性を調べる必要もないので、例えば同時実行の可能性が否定される原因となった命令が実行完了しているか否かだけを調べて、その命令を再実行する。このような命令の再実行処理により、或るサイクル後には前記インストラクション・バッファ1にフェッチした全ての命令を実行することが可能となるので、その時点で前記インストラクション・バッファ1に新たな命令(複数の命令)をフェッチする。そしてこれらの命令について同様にして前記複数の命令実行ユニット3による並列的な命令実行を制御する。

このように構成された電子計算機によれば、例えば命令のフェッチ(命令アクセス)、命令のデコード(同時実行の可能性の検出を含む)、命令の実行をステージ分割してパイプライン処理するような場合、上述した複数の命令間の同時実行可能性の判断を上述したデコードステージで実行することが可能となる。つまり命令間の同時実行可能性の判断に要する処理時間を改めて設定するこ

は、このようにして同時実行可能性検出部5にて同時実行不可能であると判断された命令の実行を中止させるべく、その命令が供給された命令実行ユニット3に対してアポート信号を発生する。同時実行の可能性が否定された命令に対して免せられるアポート信号をどの命令実行ユニット3に与えるかは、その命令の種別に応じて前記命令分配器2と同様に、その命令をどの命令実行ユニット3に分配するか(或いは分配されているか)を調べることによって決定される。命令実行ユニット3は、このようにしてアポート信号が与えられたとき、前記命令分配器2を介して与えられた命令の実行を取り止める。

さてこのようにして或る命令の実行が取り止められた場合、前記インストラクション・バッファ1への新しい命令(複数の命令)のフェッチが停止される。そして次のサイクルでは、先に取り止めとなった命令の再実行が行われる。この場合、既に実行を開始した命令については再実行する必要はない。またこの場合には既に実行完了となっ

となく、デコードステージの処理時間内に上記同時実行可能性判断の処理を行うことができる。従って複数の命令を並列的に実行する場合であっても、わざわざその為の処理ステージを設定する必要がないので、高速度に効率良く複数の命令を並列処理することが可能となる。

つまり本発明によれば、レジスタをアクセスする為に各命令実行ユニットが持つレジスタフィールドのデコードとは別個に、複数の命令の同時実行の可能性を判断する為のレジスタフィールドデコードを設け、このレジスタフィールドデコードを用いて命令分配器2とは独立に命令の同時実行可能性を判断して複数の命令実行ユニットによる命令の同時実行を制御するので、命令の同時実行可能性の判断自体を高速度に行うことができ、またそのサイクルタイムや処理実行時間の短縮化を図って計算機性能を大幅に向上させることができる。しかも命令分配器2により複数の命令を複数の命令実行ユニットに分配しながら、同時実行可能性の判断結果に従ってその命令の同時実行を制御す

るので、従来のシーケンシャルな命令コードとそのオブジェクトとの互換性を保ちながら、その命令を効率的に並列処理することを可能とする等の実用上多大なる効果を奏する。

尚、本発明は上述した実施例に限定されるものではない。実施例では4つの命令を並列的に処理する場合を例に説明したが、その数は幾つであっても良い。但し、その命令数に応じて、それらの命令間の同時実行可能性（命令実行順序）をそれぞれ検定することが必要である。また命令実行ユニットの数やその種類についても計算機の仕様に応じて定めれば良いものである。その他、本発明はその要旨を逸脱しない範囲で種々変形して実施することができる。

【発明の効果】

以上説明したように本発明によれば、シーケンシャルな命令コードとオブジェクトとのバイナリコンパチビリティ（互換性）を保ちながら複数の命令に対する並列的な処理を行うことが可能であり、その処理実行時間の短縮化を図って効率良

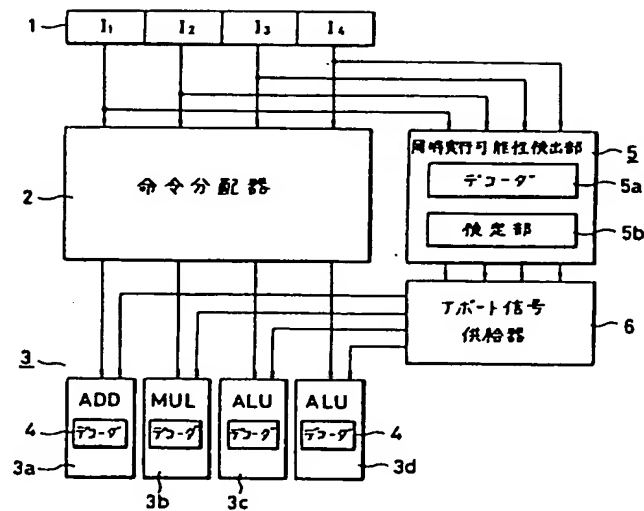
く命令を実行することができる等の実用上多大なる効果が奏せられる。

4. 図面の簡単な説明

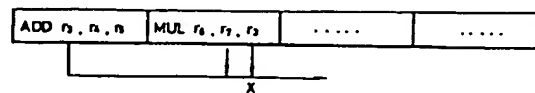
第1図は本発明の一実施例に係る電子計算機の要部概略構成図、第2図は複数の命令に対する同時実行可能性判断処理の基本的な概念を模式的に示す図である。

1…インストラクション・バッファ（フェッチ回路）、2…命令分配器、3…命令実行ユニット、4…デコード、5…同時実行可能性検出部（第1の処理ユニット）、5a…デコード、5b…検定部、6…アポート信号供給器（第2の処理ユニット）。

出願人代理人 弁理士 鈴江武彦



第1図



第2図

【公報種別】特許法第17条の2の規定による補正の掲載

【部門区分】第6部門第3区分

【発行日】平成10年(1998)12月22日

【公開番号】特開平4-54638

【公開日】平成4年(1992)2月21日

【年通号数】公開特許公報4-547

【出願番号】特願平2-164552

【国際特許分類第6版】

G06F 9/38 370

380

【FI】

G06F 9/38 370 C

380 A

手 続 補 正 書

平成 9 年 6 月 23 日

特許庁長官 荒井 寿光 殿

1. 事件の表示

特願平2-164552号

2. 発明の名称

電子計算機

3. 補正をする者

事件との関係

特許出願人

(307) 株式会社 東 芝

4. 代理人

東京都千代田区霞が関3丁目7番2号

特許事務所 内外閣特許事務所内

〒100 電話03(3502)3181(大代表)

(5847) 弁護士 鈴 江 成 彦



5. 自発補正

6. 補正対象書類名

(1) 明細書

7. 補正対象項目名

(1) 特許請求の範囲

(2) 発明の詳細な説明

8. 補正の内容

(1) 特許請求の範囲の記載を以下の通りに訂正する。

(2) 明細書第6頁第7行～第7頁第2行の記載を下記の通り訂正する。

記

本発明に係る電子計算機は、複数の命令をフェッチするフェッチ手段と、前記フェッチ手段によりフェッチされた複数の命令を、複数の実行手段にそれぞれ供給する処理を行う命令供給手段と、前記命令供給手段による供給する処理に並行して、前記フェッチ手段によりフェッチされた複数の命令について前記各命令間の実行順序の依存性を検出する第1の処理手段とを具備したことを特徴とする。

好ましくは、前記第1の処理手段により検出された複数の命令間の実行順序の依存性に従って、前記実行手段に供給された命令をアボート制御する第2の処理手段をさらに具備するようにしてもよい。

(3) 明細書第7頁第4行～同頁第18行にかけて「フェッチ回路にフェッチされた…複数の実行ユニットにおける命令の実行が制御される」とあるのを「フェッチ手段によりフェッチされた複数の命令を、その命令の種類に応じて複数の実行手段にそれぞれ供給する処理に並行して、第1の処理手段にて前記フェッチ手段によりフェッチされた複数の命令について各命令間の実行順序の依存性を検出する処理を同時に実行し、この検出結果に従って、複数の実行手段における命令の実行が制御される」と訂正する。

(4) 明細書第7頁第16行に「実行ユニット」とあるのを「実行手段」と訂正する。

(5) 明細書第7頁第18行に「実行ユニット」とあるのを「実行手段」と訂正する。



(2)

1

2

2. 特許請求の範囲

(1) 複数の命令をフェッチするフェッチ手段と、前記フェッチ手段によりフェッチされた複数の命令を、複数の実行手段にそれぞれ供給する処理を行う命令供給手段と、前記命令供給手段による供給する処理に並行して、前記フェッチ手段によりフェッチされた複数の命令について前記各命令間の実行順序の依存性を検出する第1の処理手段とを具備したことを特徴とする電子計算機。(2) 前記第1の処理手段により検出された複数の命令間の実行順序の依存性に従って、前記実行手段に供給された命令をアボート制御する第2の処理手段をさらに具備したことを特徴とする請求項1に記載の電子計算機。

(19) Japan Patent Office (JP)

(11) Publication of Patent Application

Hei 4 - 54638

(12) Publication of Japanese Laid-Open Patent Application (A)

(43) Date of Publication: February 21, 1992

(51) Int. Cl ⁵	Class. Symbol	Internal No.
G 06 F 9/38	370 C	7927 - 5B
	380 A	7927 - 5B
Request for Examination: Not requested		
Number of Inventions: 2		

(Total of 5 pages in the original)

(54) Title of the Invention:

ELECTRONIC COMPUTER

(21) Japanese Patent Application: Hei 2 - 164552

(22) Filing date: June 22, 1990

- (72) Inventor: Kenji MINAGAWA
Toshiba Research Institute (Currently, Toshiba Corporate
Research & Development Center)
1, Komukaitoshiba-cho, Saiwai-ku, Kawasaki-Shi,
Kanagawa-ken
- (72) Inventor: Ken [or Takeshi] AIKAWA
Toshiba Research Institute (Currently, Toshiba Corporate
Research & Development Center)
1, Komukaitoshiba-cho, Saiwai-ku, Kawasaki-Shi,
Kanagawa-ken
- (72) Inventor: Mitsuo SAITO
Toshiba Research Institute (Currently, Toshiba Corporate
Research & Development Center)
1, Komukaitoshiba-cho, Saiwai-ku, Kawasaki-Shi,
Kanagawa-ken
- (71) Applicant: Toshiba Corporation
72, Horikawa-cho, Saiwai-ku, Kawasaki-shi, Kanagawa-ken
- (74) Agent: Patent agent: Takehiko SUZUE and three others

SPECIFICATION

1. TITLE OF THE INVENTION:

ELECTRONIC COMPUTER

2. SCOPE OF PATENT CLAIMS:

Claim 1

An electronic computer wherein, in an electronic computer provided with a fetch circuit that simultaneously fetches multiple instructions, and an instruction distributor that distributes & supplies the multiple instructions fetched by the fetch circuit, to multiple execution units, respectively, and that allows them to execute [the instructions] in parallel, it is provided with a 1st processing unit that decodes the multiple instructions fetched by the fetch circuit, respectively, and that detects the execution order dependency between each of the instructions, parallel to the distribution & supply processing by the instruction distributor.

Claim 2

The electronic computer according to Claim 1, wherein, it is provided with a 2nd processing unit that controls to abort the instructions, which have been supplied by the [execution] units, in accordance with the execution order dependency detected between [each of] the multiple instructions by the 1st processing unit.

3. DETAILED DESCRIPTION OF THE INVENTION:

[Objective of the Invention]

(Industrial Application Field)

The present invention relates to an electronic computer that executes multiple instructions in parallel.

(Prior Art)

In recent years, the development of parallel processing type electronic computers, where the execution of multiple instructions in the minutely-graded level in parallel results in speeding up of the operation processing speed, has been variously attempted. In particular, recently, parallel processing type electronic computers referred to as a VLIW system, typified by, for example, "TRACE" by Multiflow Computer, Inc. and "i860" by Intel Corporation, have been developed, and have attracted some notice. However, in this type of conventional system electronic computer, there is still an essential problem such that conventional general sequential instruction codes cannot be executed.

Further, for the purpose of executing conventional sequential instruction codes in parallel while the compatibility of their object is maintained, it is necessary to test the sequentially provided possibility of the simultaneous execution of multiple instructions. However, for the purpose of conducting this test, it is necessary to increase, for example, the number of logic stages per cycle, as a result of which the cycle time must be increased.

In the meantime, as a technique for analyzing the dependency between each of the multiple instructions for the purpose of supporting the execution of instruction(s) having latency, a register scoreboard method has been known. In this register scoreboard method, as seen for example, in "M88000" by Motorola, Inc., instructions that have been fetched to an instruction buffer are analyzed (decoded) by a register field decoder, and source register numbers and destination register numbers of the instructions obtained in this analysis (decoding) processing, are provided to the register files. At the same time, these are provided to a register scoreboard circuit. The dependency between each of the multiple instructions is analyzed by the register scoreboard circuit, and the execution order of the instructions is controlled in accordance with the analytical results.

Consideration is given to analyzing the possibility of the simultaneous execution of the instructions according to the dependency between each of the multiple instructions to be processed in parallel, using multiple instruction execution units, by utilizing the register scoreboard method. For example, sequential multiple instructions are fetched to the instruction buffer; and the types of these instructions are determined by an instruction distributor, respectively. Execution units suitable for executing each of the instructions are selected in accordance with the determined results, respectively, and each of the instructions is distributed & supplied [to the selected execution unit], respectively; and executed in parallel.

Specifically, various instruction execution units, such as a floating point multiplication unit, a floating point addition unit and an arithmetic logical unit (ALU), are prepared, and instructions suitable for the instruction execution functions are distributed & supplied to the multiple instruction execution units, respectively, with each of the instructions being executed in parallel.

In this case, the dependency between each of the instructions is analyzed using the register field decoder and the register scoreboard circuit respectively incorporated in each of the instruction execution units.

Since the dependency between each of the instructions is analyzed using the register scoreboard circuit in each of the instruction execution units after the instructions are distributed & supplied to the instruction execution units via the instruction distributor, respectively, there is the problem that it takes a long time to process. In other words, there is a problem that a long time is required for testing the possibility of the simultaneous execution of multiple instructions.

(Problems Overcome by the Invention)

As described above, in the case of processing conventional general sequential instruction codes in parallel while maintaining the compatibility of their object, it is necessary to test the execution order dependency between each of multiple instructions, and to control the simultaneous execution of the multiple instructions. However, if the dependency between each of the instructions is to be analyzed using the conventional register scoreboard method as it is, after instructions are distributed & supplied to the instruction execution units, the dependency between each of the instructions is analyzed by each of the

instruction execution units, respectively, with the problem such that it takes a long time to process.

The present invention gives due regard to these circumstances, and has the objective of providing an electronic computer where the execution order dependency between each of multiple instructions is detected at high speed, thereby enabling efficient parallel processing of multiple instructions.

[Construction of the Invention]

(Problem Resolution Means)

The present invention relates to an electronic computer provided with a fetch circuit that simultaneously fetches multiple instructions, and an instruction distributor that distributes & supplies instructions fetched by this fetch circuit, to multiple execution units, respectively, and that allows the execution units to execute the instructions in parallel. It is characterized by the fact that in parallel to the distribution & supply processing of the [multiple] instructions by the instruction distributor, it is provided with:

a 1st processing unit that decodes the multiple instructions, which have been fetched by the fetch circuit, and that detects the execution order dependency between each of the multiple instructions; and

a 2nd processing unit that controls aborting the instructions supplied to the execution units, in accordance with the execution order dependency, detected between each of the multiple instructions by this 1st processing unit; and allows the multiple execution units to execute the instructions in simultaneously executable order, respectively.

(Operation)

According to the present invention, multiple instructions fetched by the fetch circuit are distributed & supplied to the multiple execution units according to the type of the instructions. Parallel to the processing of the decoding and execution of the instructions at each of the execution units, multiple instructions fetched by the fetch circuit, are directly decoded by the 1st processing unit, and processing to detect the execution order dependency between each of the multiple instructions is simultaneously executed, the

execution of the instructions by the multiple execution units being controlled in accordance with this detection results. Therefore, for example, in the case of detecting instruction(s) where the possibility of simultaneous execution is denied according to the detection processing of the execution order dependency, it is possible to promptly control aborting the execution of the instruction(s) by the execution unit(s) where the instruction(s) has been distributed & supplied, respectively. In other words, without having a temporal delay because of detecting the dependency between each of the instructions at an individual execution unit where the instruction(s) has been distributed & supplied, a simultaneously unexecutable instruction(s) is promptly detected, and the execution of the instruction(s) can be efficiently aborted.

(Embodiment)

An electronic computer relating to one embodiment of the present invention is explained hereafter, with reference to the drawings.

Fig. 1 shows a schematic construction of the primary components of the electronic computer relating to the embodiment, wherein the symbol '1' is an instruction buffer (fetch circuit) that fetches sequential instructions, multiple instructions at a time. Here, the instruction buffer 1 constructed to simultaneously fetch four instructions, I1, I2, I3 and I4, is shown.

The four instructions, I1, I2, I3 and I4 fetched by the instruction buffer 1, are distributed & supplied to multiple instruction execution units 3 (3a, 3b, ... 3n) according to the types of instructions, via an instruction distributor 2, respectively. In this embodiment, as the multiple instruction execution units 3, a floating point addition unit (ADD) 3a, a floating point multiplication unit (MUL) 3b, and arithmetic logical units (ALU) 3c & 3d are prepared. In the case that an instruction fetched by the buffer 1, is a floating point addition instruction, the instruction is supplied to the floating point addition unit (ADD) 3a by the instruction distributor 2, and in the case that an instruction is a floating point multiplication instruction, the instruction is supplied to the floating point multiplication unit (MUL) 3b. In addition, in the case that an instruction is an integral number operation instruction, the instruction is distributed & supplied to the arithmetic logical units (ALU) 3c and 3d, respectively.

Each of these instruction execution units 3 is provided with a decoder 4, respectively, and

basically, instructions that have been selectively distributed & supplied from the instruction buffer 1 are respectively decoded under the control of the instruction distributor 2, and a register is accessed in accordance with the data in the register field indicated in the decoding result, and the instructions are executed.

In the meantime, the four instructions, I1, I2, I3 and I4 fetched by the instruction buffer 1 are supplied to the instruction distributor 2, and are distributed & supplied to the instruction execution units 3, and are simultaneously provided to a simultaneous execution possibility detector (1st processing unit) 5. The simultaneous execution possibility detector 5 is provided with:

a decoder 5a, which decodes instructions provided from the instruction buffer 1, respectively, and

a detector 5b, which tests the decoding result and determines the possibility of the simultaneous execution between each of the multiple instructions. The possibility of the simultaneous execution between each of the four instructions, I1, I2, I3 and I4 fetched by the instruction buffer 1 is determined using the decoder 5a and the detector 5b.

Specifically, for the determination of the possibility of simultaneously executing the multiple instructions, for example, in the case that the instructions I1, I2, ... are provided, then:

I1 [ADD	r3,	r4,	r5]
I2 [ADD	r6,	r7,	r8]

...

For example, instruction I1 is regarded as a standard and the execution condition of the next instruction I2 is determined. In this case, since all of the register fields of the instruction I2 are different from the destination register filed r3 of the instruction I1, it is determined that the instruction I2 is simultaneously executable along with the instruction I1. Determination processing is performed by sequentially determining whether the execution of the next instruction can be realized relative to a predecessor instruction. In other words, the determination of the possibility of simultaneous execution is similarly performed to the next instruction I3 relative to the instructions I1 and I2, respectively. For the next instruction I4, the determination is similarly performed relative to the instructions I1, I2 and I3, respectively.

In the case of instructions I1, I2:

I1 [ADD r3, r4, r5]
 I2 [ADD r6, r7, r3]

...

As shown in Fig. 2, if the instruction I1 is regarded as a standard and the execution condition of the next instruction I2 is determined, the source register field r3 in the instruction I2 is detected as being consistent with the destination register field r3 in the instruction I1. As a result, the simultaneous execution possibility detector 5 determines that the instruction I2 needs to be executed after the execution of the instruction I1 is completed. In other words, it is impossible to simultaneously execute instruction I2 and instruction I1.

In order to abort the execution of instructions which have been determined as being simultaneously unexecutable by the simultaneous execution possibility detector 5, an abort signal supplier (2nd processing unit) 6 issues an abort signal to the instruction execution unit 3 to which the instruction has been supplied. The abort signal is issued to the instruction *[sic]* where the simultaneous execution possibility has been denied, and the determination about to which instruction execution unit the abort signal is provided is made as a result of a search about which instruction execution unit the instruction will be distributed (or has been distributed) according to the type of the instruction, similar to the instruction distributor 2. When the abort signal is provided to the instruction execution unit 3, as described above, the execution of the instructions are aborted which have been provided via the instruction distributor 2.

In the case that the execution of one instruction has been aborted, the fetching of new instructions (multiple instructions) by the instruction buffer 1 is terminated. In the next cycle, re-execution of the aborted instruction is performed. In this case, it is unnecessary to re-execute other instructions where execution has already commenced, in which case it is also unnecessary to check the possibility of simultaneous execution between instructions requiring execution, and other instructions where execution has already been completed. Therefore, whether the execution of only the instruction which caused the denial of the possibility of the simultaneous execution has been completed is checked, and the instruction is re-executed. Re-execution processing of the instruction enables the execution of all instructions fetched by the instruction buffer 1, and after one-cycle the instruction buffer 1 fetches new instructions (multiple instructions). Parallel execution of

these instructions by the multiple instruction execution units 3 is similarly controlled.

An electronic computer constructed as described above, for example, where fetching of instructions (instruction access), decoding of the instructions (including the detection of the possibility of the simultaneous execution) and the execution of the instructions are divided into stages, and pipeline processing is performed, making it possible to determine the possibility of simultaneous execution between each of the multiple instructions in the decoding stage. In other words, the processing of the determination for the possibility of simultaneous execution can be performed within the processing time for the decoding stage, without re-setting the processing time required for the determination of the possibility of simultaneous execution between each of the instructions. Therefore, even in the case of executing multiple instructions in parallel, it is unnecessary to establish the processing stage only for this purpose, making it possible to efficiently perform the processing of multiple instructions in parallel at high speed.

In other words, according to the present invention, separating the register field included in each instruction execution unit from the decoder for the purpose of accessing the register, a register field decoder for the purpose of determining the possibility of the simultaneous execution between each of multiple instructions is established, and the possibility of the simultaneous execution between each of the instructions is determined, and the simultaneous execution of the instructions by the multiple instruction execution units is controlled using the register field decoder, independent from the instruction distributor 2. Therefore determination of the possibility of simultaneous execution between each of instructions can be performed at high speed, the cycle time and the processing execution time can be reduced, and the computer's performance can be drastically improved. In addition, while the instruction distributor 2 distributes multiple instructions to the multiple instruction execution units, simultaneous execution of the instructions can be controlled in accordance with the determination results relating to the possibility of simultaneous execution. Hence, it is greatly effective in practical use; for example, and while maintaining the compatibility between the conventional sequential instruction codes and their objects, it enables the efficient parallel processing of the instructions.

However, the present invention is not limited to this embodiment. For example, an embodiment has been explained using an instance where four instructions are processed in parallel. However, this could be any number. However, it is necessary to test the possibility of the simultaneous execution (instruction execution order) between each of the

instructions, respectively. Further, the number of the instruction execution units and their type can be established according to the specifications of the computer. Other than that, the present invention can be implemented with various modifications within the scope of the outline.

[Efficacy of the Invention]

As described above, according to the present invention, while a 'binary compatibility' (compatibility [*sic.*]) between the sequential instruction codes and their object is maintained, it is possible to perform parallel processing of multiple instructions, and is greatly effective for the practical use. For example, the reduction of the processing execution time enables the efficient execution of instructions.

4. BRIEF DESCRIPTION OF DRAWINGS:

Fig. 1 is a schematic block diagram of primary components in the electronic computer relating to one embodiment of the present invention, and

Fig. 2 is diagram that shows a basic concept of the determination processing of the simultaneous execution possibility to multiple instructions as a pattern.

1 ... instruction buffer (fetch circuit), 2 ... instruction distributor, 3 ... instruction execution unit, 4 ... decoder, 5 ... simultaneous execution possibility detector (1st processing unit), 5a ... decoder, 5b ... detector, 6 ... abort signal supplier (2nd processing unit)

Agent of patent applicant:

Patent agent:

Takehiko SUZUE

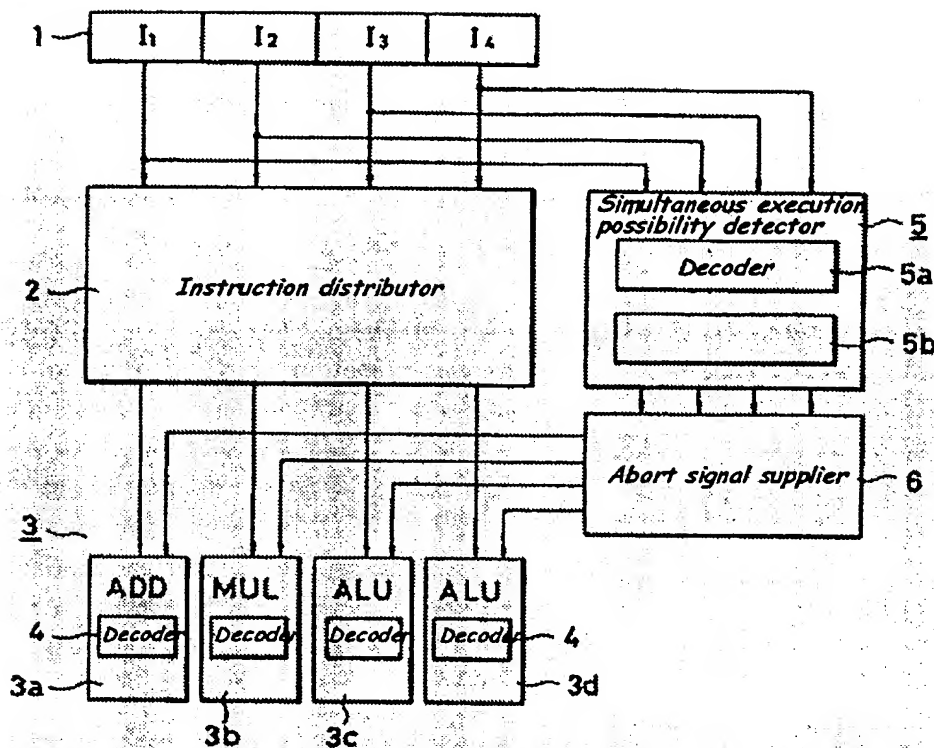


FIG. 1

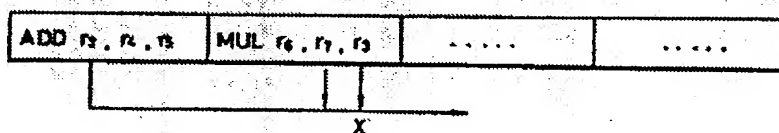


FIG. 2

BEST AVAILABLE COPY

[Publication Classification] Publication of amendment according to the stipulation in
Patent Law Article 17 Section 2

[Section Division] 6th Section, 3rd Division

[Issued Date] December 22, 1998

[Laid-Open Number] Japanese Laid-Open Patent Application Hei 4 – 54638

[Publication Date] February 21, 1992

[Serial Number] Publication of Japanese Laid-Open Patent Application 4 – 547

[Application Number] Japanese Patent Application 2 – 164552

[International Patent Type: 6th Version]

G 06 F	9/38	370
		380

[FI]

G 06 F	9/38	370	C
		380	A

Procedural Amendment

June 23, 1997

Commissioner of Patent Office: Mr. Juko [or Hisamitsu] ARAI

1. Indication of Case:

Japanese Patent Application Hei 2 – 164552
2. Title of the Invention:

ELECTRONIC COMPUTER
3. Amender:

Relationship with the Case: Patent applicant
(307) Toshiba Corporation
4. Agent:

(5847) Takehiko SUZUE

Suzuei Domestic & International Patent Firm

7 – 2, Kasumigaseki 3-chome, Chiyoda-ku, Tokyo 100

TEL: 03 (3502) 3181 (key number)
5. Voluntary Amendment:
6. Document Name Subject for Amendment:

(1) Specification
7. Item Name Subject for Amendment

(1) Scope of Patent Claims

(2) Detailed Description of the Invention
8. Contents of Amendment:

- (1) The descriptions of “Scope of Patent Claims” are corrected as in the separate paper.
- (2) The description from the 7th line of the 6th page to the 2nd line of the 7th page in the Specification is corrected as follows:

Record

The electronic computer relating to the present invention is characterized by the fact that it is provided with:

a fetch means that fetches multiple instructions;

an instruction supply means that performs the processing to supply the multiple instructions fetched by the fetch means, to multiple execution means; and

a 1st processing means that detects the dependency of the execution order between each of the multiple instructions fetched by the fetch means, parallel to the processing to supply [the multiple instructions] by the instruction supply means.

Preferably, it can be additionally provided with a 2nd processing means that controls aborting the instructions supplied to the execution means, in accordance with the dependency of the execution order detected between each of the multiple instructions by the 1st processing means.

- (3) From the 14th lines of the 7th page to the 13th line of the 7th page [*sic.*] in the Specification, the description, “fetched by the fetch circuit ... the execution of the instructions by the multiple execution units is controlled”, is corrected to, “parallel to the processing to supply multiple instructions fetched by the fetch circuit, to multiple execution means according to the types of the instructions, respectively, the processing to detect the dependency of the execution order between each of the multiple instructions fetched by the fetched means, is simultaneously executed by the 1st processing means, and the instruction execution by the multiple execution means is controlled in accordance with the detection results.”
- (4) The term, “execution unit” in the 16th line of the 7th page [*sic.*] in the Specification is corrected to “execution means”.
- (5) The term, “execution unit” in the 18th line of the 7th page [*sic.*] in the Specification is corrected to “execution means”.

2. SCOPE OF PATENT CLAIMS:

Claim 1

An electronic computer, wherein, it is provided with

a fetch means that simultaneously fetches multiple instructions,
an instruction supply means that performs the processing to supply the multiple instructions, which have been fetched by the fetch means, to multiple execution means, respectively, and
a 1st processing means that detects the execution order dependency between each of the multiple instructions fetched by the fetch means, and accomplishes parallel processing to supply [the instructions] by the instruction supply means.

Claim 2

The electronic computer according to Claim 1, wherein, it is additionally provided with a 2nd processing means that controls aborting the instructions supplied to the execution means, in accordance with the execution order dependency detected between [each of] the multiple instructions by the 1st processing means.